Published online 6 November 2020

NAR Genomics and Bioinformatics, 2020, Vol. 2, No. 4 **1**
doi: 10.1093/nargab/lqaa092

Accelerating phylogeny-aware alignment with indel evolution using short time Fourier transform

Massimo Maiolo [1,2,*], Simone Ulzega[1,2], Manuel Gil[1,2,†] and Maria Anisimova[1,2,†]

[1]Institute of Applied Simulation, School of Life Sciences and Facility Management, Zurich University of Applied Sciences (ZHAW), CH-8820 Wädenswil, Switzerland and [2]Swiss Institute of Bioinformatics (SIB), CH-1015 Lausanne, Switzerland

Received June 23, 2020; Revised October 15, 2020; Editorial Decision October 19, 2020; Accepted October 22, 2020

ABSTRACT

Recently we presented a frequentist dynamic programming (DP) approach for multiple sequence alignment based on the explicit model of indel evolution Poisson Indel Process (PIP). This phylogeny-aware approach produces evolutionary meaningful gap patterns and is robust to the 'over-alignment' bias. Despite linear time complexity for the computation of marginal likelihoods, the overall method's complexity is cubic in sequence length. Inspired by the popular aligner MAFFT, we propose a new technique to accelerate the evolutionary indel based alignment. Amino acid sequences are converted to sequences representing their physicochemical properties, and homologous blocks are identified by multi-scale short-time Fourier transform. Three three-dimensional DP matrices are then created under PIP, with homologous blocks defining sparse structures where most cells are excluded from the calculations. The homologous blocks are connected through intermediate 'linking blocks'. The homologous and linking blocks are aligned under PIP as independent DP sub-matrices and their tracebacks merged to yield the final alignment. The new algorithm can largely profit from parallel computing, yielding a theoretical speed-up estimated to be proportional to the cubic power of the number of sub-blocks in the DP matrices. We compare the new method to the original PIP approach and demonstrate it on real data.

INTRODUCTION

Today's large genomics datasets provide a rich source of information and enable increasingly realistic models to be applied to study the underlying mechanisms shaping biological sequences. Such models however tend to be mathematically more sophisticated, and as a consequence, are computationally more demanding. In this context, one of the oldest and most fundamental problems is the alignment of related genomic sequences. This problem is well-known in the bioinformatics community as multiple sequence alignment (MSA).

Due to the inherent computational complexity of the MSA inference, heuristic algorithms have been developed to enable this task as a part of routine sequence analyses. The progressive MSA heuristics simplify the problem by splitting it into a series of pairwise alignments guided by a tree structure representing the evolutionary relationship of the sequences. Each pairwise alignment is typically constructed by dynamic programming (DP), which usually scales quadratically with the sequence length. A typical approach however considers only point substitutions and a length distribution of observed sequence gaps. Including more sophisticated scenarios necessitates methods of higher complexity. For example, the computational complexity of a pairwise alignment with non-overlapping inversions becomes cubic with the sequence length (1).

A sound mathematical description of the evolutionary process of insertions and deletions (indels) requires more complex models such as the classical TKF91 (2) or the more recent Poisson Indel Process (PIP) (3). The advantage of both models consists in describing the evolution of indels on a tree. Their computational complexity is largely determined by the evaluation of the marginal likelihood of an MSA and a tree, which is exponential in the number of taxa for TKF91, but is reduced to linear for PIP. The Poisson Indel Process is a mathematical model that describes an evolutionary process of character substitutions, deletions and insertions along a phylogenetic tree. Single character insertions occur over time as poissonian events, and the inserted characters evolve under a continuous-time Markov process of substitutions and deletions. Once a character is deleted, its homology history expires in order to prevent a subsequent insertion. The gap patterns of an MSA generated un-

*To whom correspondence should be addressed. Tel: +41 58 934 53 45; Fax: +41 58 934 50 01; Email: massimo.maiolo@zhaw.ch
†The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors.
Disclaimer: The funding body did not play any role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

der PIP are defined using two parameters, insertion rate and deletion rate.

Recently we have presented a new progressive DP algorithm (4) that aligns two MSAs under the PIP model by maximum likelihood (ML). This algorithm runs through a given guide tree and computes at each internal node a column-wise likelihood for all the homology paths implied by the two sub-alignments observed at the children nodes. The DP algorithm then returns the optimal ML pairwise alignment conditioned on the input. However, this approach requires sparse three-dimensional (3D) DP matrices to account for the non-monotonicity of the marginal likelihood for non-observable scenarios. As a consequence, the overall computational complexity of the method becomes cubic in the sequence length.

A possible approach to reducing computational complexity in a DP framework is to pre-detect candidate homologous sequence segments in order to filter out non-promising regions in the DP matrix prior to the effective alignment process (5,6). This allows to heavily constrain the number of candidate alignments thus reducing the overall problem complexity. One of the fastest and accurate popular aligners, MAFFT (6,7), relies on a fast Fourier transform (FFT) for the homologous segments detection. The usage of FFT for alignment can be traced back to the work by Felsenstein (8) who applied it to obtain ungapped pairwise alignments of entire sequences in $\mathcal{O}(L \log L)$ (where L is the average sequence length). Nonetheless, the method was deemed of 'limited value' because of the impossibility of accommodating indels. In MAFFT ungapped homologous segments are used to constrain possible DP paths and, thus, exclude areas from the DP calculation. Gappy regions then link consecutive homologous regions thus yielding the final MSA. The resulting speed up increases with an increasing number of detected homologous segments.

Here, we present a novel FFT-inspired approach to identify homologous segments and apply it in the progressive DP-PIP framework. Our approach differs from MAFFT in several aspects: (i) instead of FFT, we use a multiple-resolution short-time Fourier Transform (STFT), which improves the detection of homologous regions especially for distantly related sequences; (ii) we define a general approach to construct logically sound paths connecting homologous blocks and to resolve overlaps between them; (iii) we compute several critical tuning parameters directly from the input data, instead of relying on hard-coded default values.

In addition, the proposed method is easily parallelized. Finally, we show that compared to the original DP-PIP algorithm the STFT approach produces very similar alignments. The new method is demonstrated on real data.

MATERIALS AND METHODS

Converting amino acid sequences to signals of physicochemical properties

Substitutions between amino acids with similar physicochemical properties are known to be more frequent than those between chemically distant ones. Replacements by similar amino acids tend to preserve the structure of proteins and are, therefore, more likely to occur during evolution. Based on this, MAFFT (6) detects presumed homologous amino acids between evolutionary related sequences with a cross-correlation-based analysis of their physicochemical properties, namely *volume* and *polarity*. The degree of cross-correlation acts as a measure of physicochemical similarity and, thus, serves as proxy for the likelihood that the sequences might undergo substitutions. Here, in addition to volume and polarity, we also consider chemical composition, as suggested in (9). Consequently, in our method an amino acid sequence of length L is represented as a $3 \times L$ matrix $\mathbf{s}$, subsequently referred to as *signal*,

$$\mathbf{s} = \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_L \\ p_1 & p_2 & p_3 & \dots & p_L \\ c_1 & c_2 & c_3 & \dots & c_L \end{bmatrix}, \tag{1}$$

where v_i, p_i and c_i, $i = 1, \dots, L$, denote *volume*, *polarity* and *chemical composition*, respectively, of the i-th amino acid in the sequence (9). More generally, to convert an MSA into a multi-dimensional signal $\mathbf{s}$, we define v_i, p_i and c_i as the average volume, polarity and chemical composition, respectively, of the amino acids aligned in the i-th column of the MSA. Since the magnitude of the three physicochemical properties $\mathbf{v}$, $\mathbf{p}$, $\mathbf{c}$ varies significantly, MAFFT standardizes the signal assuming a homogeneous distribution of the 20 amino acids. We have refined the method to allow for non-homogeneous distributions. Therefore, a standardized volume $\widehat{\mathbf{v}}$ is defined by $\widehat{\mathbf{v}} = (\mathbf{v} - \overline{v})/\sigma_v$, where $\overline{v}$ and σv are the sample mean and standard deviation over the L volume components in the data. Analogous definitions hold for $\widehat{\mathbf{p}}$ and $\widehat{\mathbf{c}}$. We refer to the standardized signal as $\widehat{\mathbf{s}}$.

Computing cross-correlation of physicochemical properties signals

Given two MSAs represented by signals $\widehat{\mathbf{s}}_1$ and $\widehat{\mathbf{s}}_2$, their cross-correlation returns a discrete function $f[k]$, where k is the relative positional shift between $\widehat{\mathbf{s}}_1$ and $\widehat{\mathbf{s}}_2$:

$$f[k] = \sum_{1 \le i, i+k \le L} \widehat{\mathbf{s}}_{1,i} \cdot \widehat{\mathbf{s}}_{2,i+k} \,. \tag{2}$$

The product in the sum denotes the scalar product between the column vectors indexed by i and $i + k$.

Peaks in the function $f[k]$ identify shifts k for which regions in the two mutually shifted sequences show a high degree of similarity and, therefore, evidence for putative homology. Note that the signals $\widehat{\mathbf{s}}_1$ and $\widehat{\mathbf{s}}_2$ can be padded and therefore do not need to be of equal length (see Appendix Signal padding).

The cross-correlation operation (Equation 2) can be rewritten in terms of the Fourier Transform (see Appendix The Fourier transform). An FFT algorithm (10) reduces the computational complexity of the cross-correlation from $\mathcal{O}(L^2)$ to $\mathcal{O}(L \log L)$, where L is the average sequence length. Moreover, the FFT approach further reduces the original $\mathcal{O}(L^2)$ complexity required by a classic DP based aligner by filtering out the non-promising regions in the DP matrix. This last reduction is a function of the number of homologous blocks and their sizes (6).

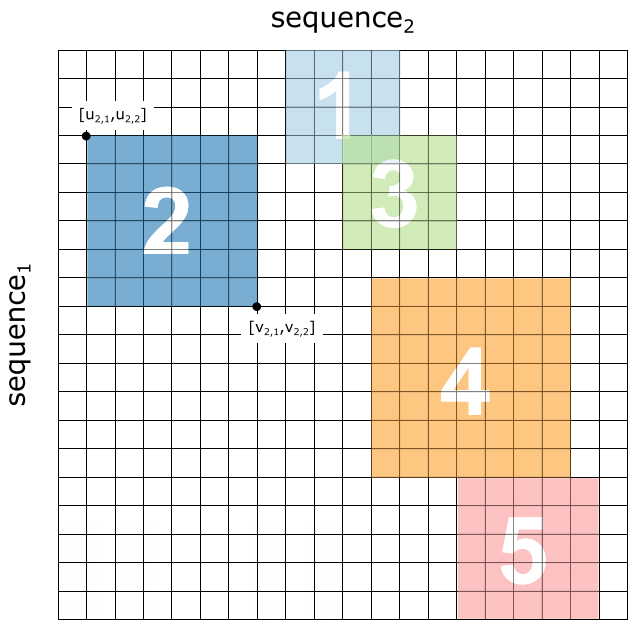

Figure 1. An hypothetical example of a homology matrix with detected homology blocks. Shown are five homologous blocks with various forms of overlap. Each block is defined by the coordinates of the upper-left [$u_{j,1}$, $u_{j,2}$] and bottom-right [$v_{j,1}$, $v_{j,2}$] vertices, e.g. as shown for block 2.

Homologous block localization

As pointed out above, the cross correlation measures the similarity $f[k]$ between two sequences, as a function of the shift k of one relative to the other. Values of k for which $f[k]$ is high, indicate that the shifted sequences contain overlapping regions with high similarity. However, the actual locations of the regions on the sequences are not provided. MAFFT localizes homologous regions by computing column-wise scores on the shifted sequences. Columns with a score above a threshold of 0.7 are deemed to be homologous. Note that the threshold is hard coded.

Instead, we propose to use a multiple-resolution STFT (11), which applies a series of Fourier Transforms to a moving windowed signal, thus enabling a precise localization of putative homologies. Therefore, the STFT provides simultaneous information on both the shift k and the location of resembling residue patterns. The STFT analysis is performed at increasing levels of resolution. The gradual reduction of the size of the moving window in which the signal is analyzed yields a progressively more precise localization of the detected similar blocks, albeit at the greater computational cost.

The putative homologous regions detected by the STFT define a set of blocks in the MSA homology matrix, as shown for example in Figure 1. Such blocks are then selected and linked so as to maximize the coverage of the homology matrix. The selected blocks are then aligned independently and assembled to build the final MSA. Note that the selection of blocks helps to reduce the number of cells that have to be computed by the DP procedure. This is particularly effective for full ML alignment under the explicit indel model PIP (4), where the homology matrix is 3D.

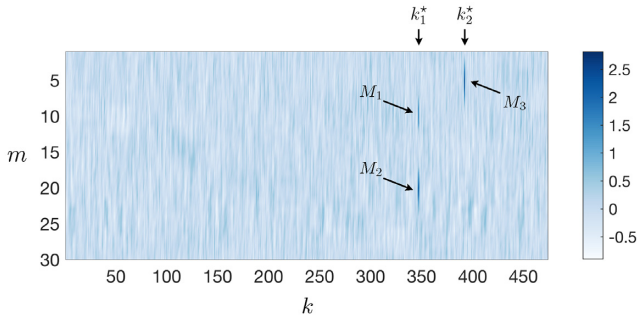

Figure 2. Graphical representation of a full cross-correlation matrix $fw[m, k]$ generated using two arbitrary synthetic sequences of 1000 amino acids. One can identify two distinct shifts $k_1^\star$ and $k_2^\star$ where the cross-correlation exceeds the noise threshold t_h, as explained in the text. The condition is fulfilled by the sets of window displacements M_1 and M_2, when $k = k_1^\star$, and by the set M_3 when $k = k_2^\star$.

In the following sections, we describe in more detail the STFT-based algorithm for block detection, selection and linking.

STFT-based algorithm

As mentioned above, MAFFT localizes potential homologous regions based on column-wise scores. These scores tend to be high for virtually gap-free columns and, conversely, very low for columns with high gap content. This approach, however, cannot be applied under the PIP model, where there is no correlation between a column likelihood and its gap content, for example see Supplementary Figure D.1.

Instead, in our method regions of high similarity are localized using a multi-scale STFT analysis (Appendix The multi-scale STFT). For a given window function ψ of size w, the cross-correlation is a 2D matrix $fw[m, k]$. It is a discrete function of both the shift k and the location m of a moving window on the shifted sequences (see Figure 2 and Appendix The multi-scale Short-Time Fourier Transform STFT). We start with a moving window of large support size w to detect the shifts k, which predict putative homologous regions on the shifted sequences. In this first iteration, the boundaries of the homologous regions are identified with relatively low accuracy (depending on the window size w), but at a modest computational cost. After the first coarse estimation, we progressively half both the window support w and the step size Δm. This allows us to determine the edges of the homologous regions with correspondingly higher precision. In principle, the process can be iterated until w is equal to 1. However, this is not recommended due to an increased false positive rate for shorter window sizes. Practically, we observed that two to three iterations, corresponding to a window size reduction of a factor four−eight, are sufficient to obtain a satisfactory resolution at the boundaries of the homologous patterns.

The step size Δm of the window function ψ must be such that the windows overlap each other in order to avoid numerical artifacts (12). Note that the choice of the step size affects the dimensions of the matrix $fw[m, k]$ and therefore its computational cost. Indeed, the matrix fw has a size of

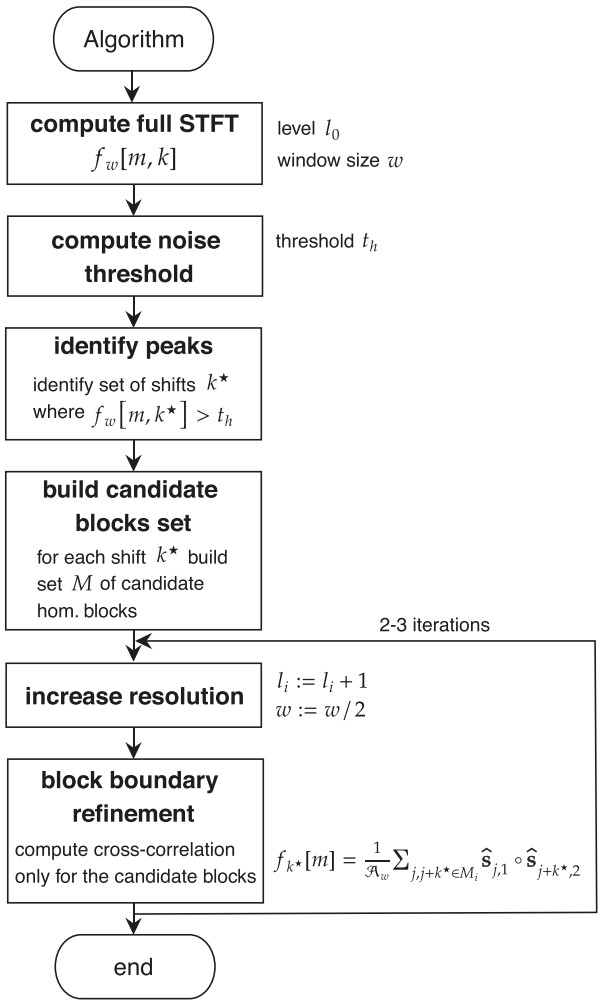

Figure 3. Algorithm scheme.

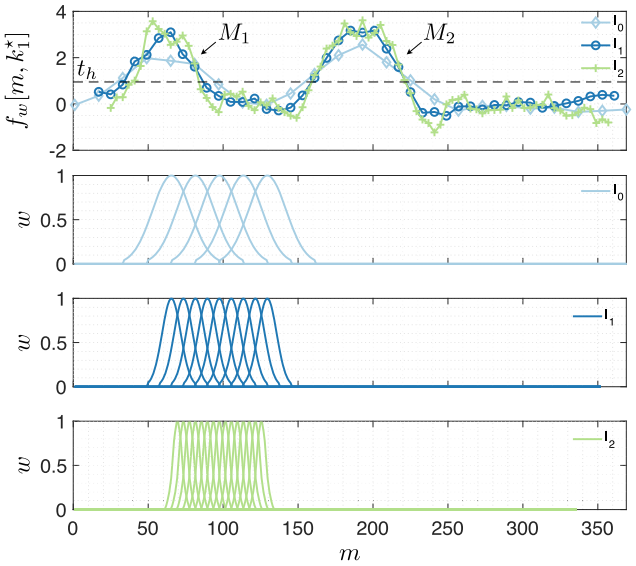

Figure 4. Boundary analysis at different levels of the multi-scale STFT algorithm. The boundaries of the blocks are analyzed at different scales. At the next level (at higher resolution) the analysis is performed only in the neighborhood of the boundary. Top: Slice of the spectrogram $fw[m, k]$ at $k_1^\star$ analyzed with a window of size 64 (l_0), 32 (l_1) and 16 (l_2).

$((\max(L_1, L_2) + w)/\Delta m) \times (\max(L_1, L_2) + w)$, where L_1 and L_2 are the lengths of the two sequences.

The main steps of the algorithm are sketched in Figure 3 and described below in details. The algorithm takes as input two sequences (signals) $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$, a window function ψ with size w and the step Δm.

i. Starting at the lowest resolution or level l_0, corresponding to the largest window support size w, e.g. 128 amino acids, we construct the full cross-correlation matrix $fw[m, k]$, as described in Equation B.2. An example cross-correlation matrix is shown in Figure 2.
ii. At level l_0 we compute a noise threshold t_h (the dashed line in Figure 4). It allows us to distinguish shifts with no similarity between the sequences from shifts containing putative homology. The threshold is calculated by recomputing the cross-correlation matrix $fw[m, k]$ after randomly permuting the residues of one of the two sequences. This procedure statistically destroys any potential homologous patterns in the sequences and allows us to define an intrinsic 'noise' level as the maximum of the cross-correlation coefficients $fw[m, k]$. This operation is repeated until the noise level reaches a stationary value. Typically, only a few iterations are sufficient for a good estimation of the noise threshold t_h. The noise level is computed only once at the first resolution level l_0 and assumed constant in all further steps.
iii. We scan the cross-correlation matrix to identify all shifts $k^\star$ that exhibit peaks where $fw[m, k^\star] > t_h$. These peaks are clearly visible in Figure 2. Further, Supplementary Figure B.1 shows how those peaks are refined at different resolution levels, as described below.
iv. For each shift $k^\star$, the condition at point 3 is fulfilled by one or more window displacements m. The latter tend to cluster at specific locations in the matrix forming sets M_i with $i = 1, \ldots, N$. Each set M_i corresponds to a specific pattern in the sequences characterized by a high correlation coefficient, i.e. a candidate homologous block. In Figure 2, for example, one can identify three sets M_i ($i = 1, 2, 3$) of window displacements corresponding to two shifts $k_1^\star$ and $k_2^\star$. Figure 4 shows the profiles of the same putative homologous blocks.
v. We increase the resolution level from l_0 to l_1, by halving the window size w and the step size Δm.
vi. For each $k^\star$ and for each set M_i, we calculate the corresponding correlation coefficient using a slightly modified form of Equation (2),

$$f_{k^\star}[m] = \frac{1}{\mathcal{A}_w} \sum_{j,\, j+k^\star \in M_i} \hat{\mathbf{s}}_{j,1} \cdot \hat{\mathbf{s}}_{j+k^\star,2} \,. \qquad (3)$$

This way, we restrict our analysis only to the regions of interest (e.g. M_1, M_2 and M_3 in Figure 2), while most part of the cross-correlation matrix is neglected. In order to guarantee a scale-invariant analysis through the

iterative process described here, the cross-correlation function is normalized by the window area $\mathcal{A}_w = \sum_i w_i$.

vii. The one-dimensional cross-correlation of Equation (3) allows us to refine the boundaries of the candidate homologous regions M_i (e.g. Figure 4), so that they fulfil the condition $f_{k^*}[m] > t_h$.

viii. The algorithm iterates the procedure from point five to seven, increasing the resolution level to l_2, l_3 and so on. Typically, a resolution l_2 is sufficient to achieve good accuracy in estimating the edges of the individual homologous blocks. A comparison of three different resolution levels is shown in Supplementary Figure B.1.

Selection, connection and alignment of putative homologous regions

Once putative homologous regions between two MSAs are identified, the next task is to assemble an alignment based on the homologous blocks detected by STFT. To accomplish this, the required three steps are as follows: (i) select an optimal path connecting the homologous blocks (or a selection of them) in the homology matrix (see Figure 1), (ii) resolve possible overlaps between the selected blocks and align each block independently by DP, (iii) identify 'linking' blocks connecting the aligned homologous blocks, align each linking block independently, and obtain the final alignment by joining the tracebacks of all sub-alignments for homologous and linking blocks. We now describe in detail the steps outlined above.

Selecting homologous blocks and the optimal path

The optimal connecting path maximizes the total sum of involved homologous residues or, equivalently, maximize the total area of the selected blocks. This can formulated as a longest path problem through a weighted Directed Acyclic Graph (DAG), whereby the putative homologous blocks represent the nodes of the graph.

Block b_j characterized by the coordinates of its upper-left (u) and bottom-right (v) vertices, $\{[u_{j,1}, u_{j,2}], [v_{j,1}, v_{j,2}]\}$ is compatible with another block b_k if $(u_{k,1} \geq u_{j,1}) \wedge (u_{k,2} \geq u_{j,2})$. The direction associated with an edge reflects the order in which the residues appear in the sequences to be aligned. We define the edge weight as the block area of the node toward which it is directed (target). In order to search through all possible paths, the algorithm adds two extra nodes, a start and an end, defining the start and end of the optimal path. The start node is connected to all blocks and all blocks are connected to the end node. So, the weight of the start node to a given node is the area of the block it is connected to. The weight of the connection to the last dummy node is any number >0. The longest path through such weighted DAG is equivalent to the shortest path in the same DAG with negative weights and therefore it can be computed in linear time by means of the Bellman-Ford-Moore algorithm (13,14).

Overlap resolution and block alignment

The Bellman-Ford-Moore algorithm returns a list of blocks that together constitute the optimal path, maximizing the number of residues considered homologous by cross-correlation.

Since each block is treated independently by the DP alignment, the overlaps must be resolved prior to their alignment. While resolving the overlaps, the structure of the detected homologous regions should be retained as much as possible. Therefore, the overlaps are resolved by replacing the overlapping blocks by two re-sized non-overlapping blocks that retain the largest possible part of the diagonal elements of the original ones. The algorithm resolves overlaps by scrolling through the blocks and processing two of them at a time until all overlaps are removed, leaving no more residues shared by adjacent blocks.

Each resolved block is aligned as an independent submatrix by DP, resulting in a traceback path. Block diagonals correspond to matches in the DP alignment and represent therefore the expected traceback paths that we want to preserve as much as possible. To link these independent paths, we shorten each traceback path from both ends, so that it starts at the first and ends at the last match state. The aligned non-overlapping and re-sized homologous blocks are then connected by linking blocks as described below.

Linking blocks and final alignment

The linking blocks correspond to the gap-rich regions in the alignment, they join adjacent homologous blocks with virtually no gaps. The endpoint of the first path is the starting corner of the first linking block, while the starting point of the second path is the end corner of the previous linking block, and so on. Each linking block is aligned as an independent submatrix by DP, which results in linking the traceback paths. Eventually, all the tracebacks are merged to obtain the final alignment. Note that the procedure can easily be parallelized, as all blocks are independently aligned by DP. Moreover, only the selected homologous and linking blocks are aligned, while the rest of the homology matrix is excluded from the calculations.

RESULTS

To evaluate the performance of the new DP method with the STFT block detection, we compared it with: (i) our original DP approach without STFT (4), and (ii) an FT-based approach in the manner of MAFFT (6).

Alignment with and without STFT

Recall that the STFT approach reduces the input sequences to their amino acid 'signal', in order to speed up the method by pre-detecting homologous regions. Thus, the expectation is that this approach results in speed gains but does not reduce the MSA accuracy. We therefore compared the inferred MSAs for several real datasets, as listed in Table 1. For all datasets MSAs inferred with and without STFT were nearly identical. For example, Figure 5 compares the two inferred MSAs for envelope glycoprotein gp120 from HIV/SIV. As one can note from the diagonal of the matrix,

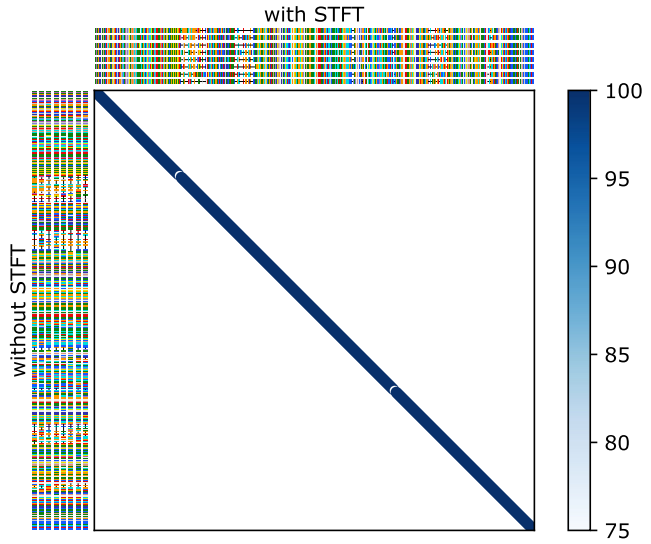

Figure 5. Two MSAs of envelope glycoprotein gp120 from eight human strains HIV inferred with STFT (top) and without the STFT (left). The degree of matching between MSAs is represented with color gradient and bubble size. The two MSAs are very similar, despite the constraints imposed by the STFT (region boundaries) and the filtering of not promising regions. Sum-of-pairs (SP) = 0.998.

Table 1. Speed-up table

	Number of DP matrix elements		
Dataset	DP	DP-STFT	Speed-up
RV912-B096*	9 016 347 960	2 800 662 444	3.2
RV913-B290*	3 184 395 957	1 869 814 875	1.7
RV913-B079*	896 083 842	177 635 637	5.0
GP120	16 240 871 307	4 568 044 458	3.6
Papillomavirus	116 033 223	45 256 197	2.6

*Datasets from BAliBASE (19).
RV912-B096: ATP-dependent DNA helicase 2 subunit KU70, five strains from human, arachnida,dictyostelia, liliopsida and oligohymenophorea. Average sequence length = 669 AA.
RV913-B290: Alpha-methylacyl-CoA racemase, nine strains form from human, amphibia, aves, actinopterygii and insecta. Average sequence length = 389 AA.
RV913-B079: Osteopontin protein, six sequences from human and mammalian. Average sequence length = 305 AA.
GP120: Envelope glycoprotein gp120 sequences from 23 strains of human and simian immunodeficiency virus. Average sequence length = 485 AA.
Papillomavirus: Protein E7 from 18 strains of human and mammalian. Average sequence length = 99 AA. More details in Supplementary Materials.

the two MSAs are almost identical, with only few differences generally coinciding with the gappy regions.

Table 1 shows the speed-up values of the aligner with and without STFT. The table reports the total number of entries to be calculated in the DP matrices in the two problems. The speed-up is then the ratio between the two numbers.

Comparison between FT and STFT

We carried out two tests to assess whether the FT or STFT approach is more effective at detecting similar patterns between two sequences.

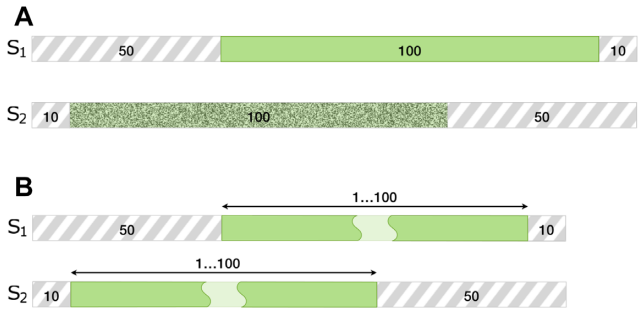

Figure 6. (**A**) Noise sensitivity experiment. The test is built in the following way: we have synthesized two amino acid sequences with a length of 160 residues. The first sequence contains a residue pattern of length 100, the second sequence is incrementally corrupted with noise by changing random residues at random positions (thereby preventing residues from being replaced by an equal residue). (**B**) Pattern length sensitivity test. We have synthesized two amino acid sequences of increasing length. The sequences have been constructed to have random residues at the head and tail, while the middle was fixed to the same pattern in both sequences. The middle pattern was incrementally extended from a length of 1 to a final length of 100 residues.

In the first test, we evaluated the method's ability to distinguish similar regions from regions with an increasing dissimilarity. In simulation studies, amino-acid substitutions are often introduced with a standard evolutionary Markovian approach. Typical amino-acid substitution models are JTT, WAG, LG and the Gonnet matrices. Such empirical models are fitted on a large number of sequences so that they capture the physicochemical properties of the amino-acids. For our evaluations, we used a random sampling process, where the substitutions take place without considering the previous state of the amino acids being replaced. The approach does not take the physicochemical properties of the residues into account. As the tested methods consider such properties, the search for similar patterns is harder than under a standard model. Specifically, we generated data by starting with two identical sequences, each of 160 residues and progressively corrupting a 100-residue pattern in one of the sequences by mutating the residue states uniformly at random (see Figure 6 A). Noise (point mutations) was introduced increasingly from 1 to 100% in steps of 1%, with three replicates per number of mutations. In accordance with terminology from signal processing, we refer to the resulting mismatches between the originally identical sequences as *noise*.

Note that the addition of noise at random does not follow any evolutionary model, and the physicochemical characteristics of the residue that is being replaced are not even taken into consideration. Our approach makes the problem more complicated and corresponds to the worst situation in biology where saturation is reached and therefore the evolutionary time reference is lost. In this test we wanted to focus only on evaluating the two approaches, namely FFT and STFT, to compare their effectiveness in detecting similar patterns. Our tests are meant to be generic, so that the results apply not only to molecular sequences separated by finite evolutionary distances, but also to any kind of generic signals. As such, our approach is also applicable in other contexts.

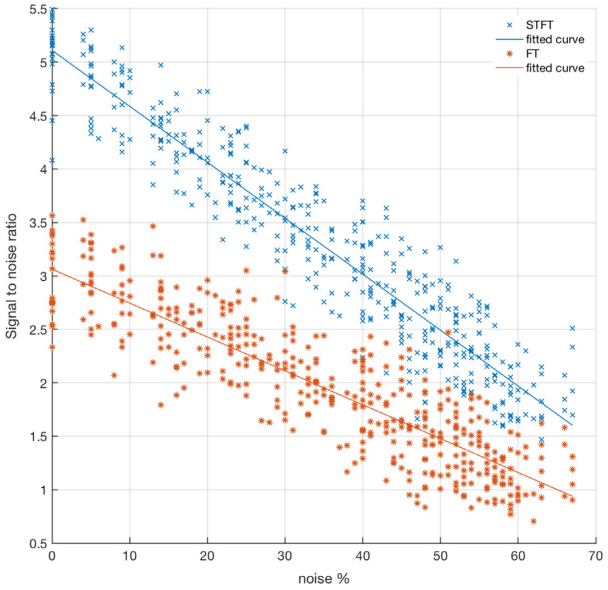

Figure 7. Noise sensitivity test: This graph shows the SNR measured between the cross correlations obtained with FT and the one obtained with STFT. We started with two sequences with a length of 100 amino acids each. Then, one of the two sequences was incrementally corrupted by noise by randomly changing residues. Each experiment was repeated several times. The figure shows that STFT has a better SNR than FT and, therefore, is more effective in detecting similar but not necessarily identical patterns.

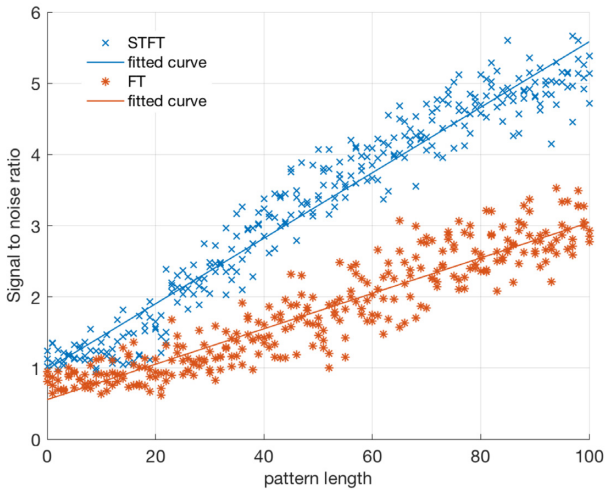

Figure 8. This graph compares the SNR obtained using FT and STFT. The values were obtained by increasing the length of an identical pattern present in both sequences. For each increment the cross-correlation and the SNR were calculated. Each experiment was repeated several times. The graph shows that STFT is more effective than FT in detecting patterns (higher SNR), also in presence of short relative patterns.

Data were generated with an increasing levels of noise, measured as a percentage of substitutions. For each new synthesized pair of sequences we have computed the cross-correlation coefficients both with the FT and the STFT approaches. We have evaluated the two MSA methods by estimating the signal-to-noise ratio (SNR), calculated as the cross-correlation value of a peak, at the known positional shift, divided by the inferred noise threshold t_h defined above. The SNR gives an indication of the ability to distinguish peaks due to highly correlated regions rather than from the noisy background. Figure 7 shows SNR values obtained with the two methods as a function of the noise content. The STFT approach is clearly more effective in finding noisy patterns compared to FT. Indeed, throughout the simulated noise range the advantage margin remained visibly large, even at the highest noise levels.

In the second test, we have quantified the ability of the method to detect short identical patterns. To this end, we synthesized two sequences of increasing length from 61 to 160 residues, which contained identical patterns of increasing length from 1 to 100 residues respectively (see Figure 6B). Once again, for each pattern length we have computed the cross-correlation coefficients both with the FT and the STFT approaches. Figure 8 shows SNR values obtained with the two methods as a function of the pattern length. We observe that the STFT approach is always more effective in finding short patterns compared to FT. The advantage margin over FT increases dramatically for longer patterns. At the same time, STFT is also better than FT at detecting very short patterns. This can be explained by the use of the window function that restricts the effectiveness of FT for a shorter segment so that the cross-correlation coefficients of similar patterns become more important.

Therefore, in both tests STFT outperformed FT with a large margin, thus supporting the application of our novel approach to homology detection.

DISCUSSION AND CONCLUSION

In this article, we described a new approach to accelerate the phylogeny-aware MSA inference based on the explicit model of indel evolution PIP. While we applied our new approach under PIP, its elements can be used more generally in a context of DP-based alignment, such as in the popular programs MAFFT (6) or PRANK (15). Although our method is inspired by MAFFT, we have introduced a number of novelties.

First, similar to MAFFT, input amino acids sequences are converted into signals representing their physicochemical properties. However, in addition to volume and polarity used in MAFFT, we included the chemical composition (described by Grantham's distance based metric), and we also standardized all the physicochemical properties, in order to allow for non-homogeneous distribution of amino acids. Note that there are several ways to define chemical composition and using a different definition may have an effect on the statistical properties of homologous blocks detection, with some definitions performing better than others.

Next, in contrast to MAFFT's FFT approach, we predetect potential homologous blocks by multi-scale STFT. The advantage is that STFT simultaneously provides information on both the positional lags and the relative positions of homologous regions. These are computed simultaneously and in a unique framework. There is therefore no need to define two different thresholds in two different and incompatible measuring systems. However, this advantage comes at a computational cost, which can be quanti-

fied for a naive implementation of STFT by a complexity of $\mathcal{O}\left(L^2 \log L\right)$, where $\mathcal{O}(\log L)$ is due to halving the window until size 1. Since we use only two or three iterations, the complexity in our case is $\mathcal{O}\left(L^2\right)$. Further, various algorithms have been proposed to reduce the complexity of STFT. For instance, for a sliding rectangular window, one can recycle most of the previously computed correlation coefficients (16–18). In addition, the sliding window can be moved in discrete steps larger than one while retaining most of the information carried by the signal. Moreover, further reductions of the computational effort can be achieved if each step of the analysis is performed only on the regions emerged at the previous coarser iteration.

Another advantage of our STFT approach is that several critical tuning parameters are computed directly from the input sequences, instead of relying on general purpose hard-coded values. For example, in our method the cardinality of the set of candidate positional lags for homologous regions is not fixed a priori but is constructed using a data-dependent and statistically robust noise threshold. In the same manner, we do not impose a fixed match-threshold to label homologous regions (set to 0.7 by MAFFT (6)), nor do we define minimum and maximum homologous block sizes, set by MAFFT to 30 and 150, respectively.

Our experiments suggest that the use of multiple-resolution STFT improves the detection of homologous regions especially for divergent sequences. Also, the use of a window function makes the STFT more effective in detecting short patterns compared to the classical FFT. Furthermore, our analyses of real data suggest that the new STFT approach does not distort the alignment accuracy, allowing to infer almost identical alignments, compared to the slower original approach.

Note that thanks to the detection of 'gappy' and 'non-gappy' blocks there is a possibility to align different regions with different model parameters, as they might evolve under different evolutionary conditions. Moreover, by labeling gappy and non-gappy blocks, in the optimization phase (e.g. MSA optimization) it is possible to focus the effort on the regions of greater variability (gappy) while keeping the rest constant. This enables an additional saving of computational time.

The homologous regions pre-detected by STFT define candidate homologous blocks within the three 3D DP matrices used under the PIP model. An optimal selection of blocks is connected through intermediate 'linking blocks'. The homologous and linking blocks are aligned under PIP as independent DP sub-matrices and their traceback paths merged to yield the final alignment. Thereby, we define a new sophisticated and general approach to generate logically sound paths to connect an optimal selection of homologous blocks and to resolve overlaps between them. This constitutes another novel and independent contribution of our work, which is applicable to other DP alignment methods.

Finally, note that due to the independence of the sub-blocks in the DP matrices, the new algorithm can largely profit from parallel computing.

It is worth mentioning that our proposed algorithm is also applicable to nucleotide sequences. We follow MAFFT's approach. Specifically, the calculations are based on the nucleotide frequencies within the sequences. The input data is converted to a $5 \times L$ matrix, having one dimension for each possible state (4 nt and one gap state). Considering that the alphabet is smaller and not exploiting Grantham's principle of substitution between residues with similar characteristics, the approach is not expected to be as accurate. In this case the method responds very well to similar patterns but cannot rely on transition patterns between nucleotides.

Today's state of the art alignment methods rely on gap costs. It is not clear how to define cost values suitable for specific datasets. Users often experiment with their data by trial and error (i.e. examining by eye the alternative MSAs produced with different gap costs), or simply resort to software defaults, which are typically defined based on empirical examination of numerous datasets. While default gap costs might work well on average, they are not directly interpretable, and there is no objective method to adapt gap costs to the data. Gap costs are not informative about the indel generating process over evolutionary time, but only describe the distribution of gap patterns in a given MSA. The progressive alignment methods with gap penalty schemes very often lead to an overestimation of deletions, a distortion called 'over-alignment'.

To avoid the over-alignment bias, PRANK keeps track of indel events on the phylogeny and adjusts gap scores in an ad-hoc manner. The lack of an explicit evolutionary indel model however, imposes significant limitations on statistical inferences. Overcoming this, PIP-based evolutionary alignment has no gap costs, but instead uses indel rates in a sound statistical context. These indel rates are not only interpretable biologically, but also can be adapted to the specific problem (i.e. optimized as model parameters). Previously, we have shown that PIP-based DP alignment avoids over-alignment, producing MSAs of similar lengths compared to PRANK (4). Via indel rate settings, the PIP-based aligner allows to infer gap patterns similar to PRANK's. These inferred gap patterns are also phylogenetically meaningful as supported by empirical studies (e.g. 20). However, aligning sequences with PIP is computationally expensive. To speed up this process we have introduced the STFT heuristics, inspired by the FFT approach introduced in MAFFT. This further enhances the MAFFT algorithm while remaining generic, so it can be combined with other DP based aligners.

DATA AVAILABILITY

The prototype implementation in MATLAB and the data used for this manuscript are available from the github repository: https://github.com/acg-team/STFT-matlab.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

FUNDING

Swiss National Science Foundation [31003A_157064, 31003A_176316 to M.A.].
Conflict of interest statement. None declared.

REFERENCES

1. Ledergerber,C. and Dessimoz,C. (2008) Alignments with non-overlapping moves, inversions and tandem duplications in O (n 4) time. *J. Comb. Chem.*, **16**, 263–278.
2. Thorne,J.L., Kishino,H. and Felsenstein,J. (1991) An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, **33**, 114–124.
3. Bouchard-Côté,A. and Jordan,M.I. (2013) Evolutionary inference via the Poisson Indel Process. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 1160–1166.
4. Maiolo,M., Zhang,X., Gil,M. and Anisimova,M. (2018) Progressive multiple sequence alignment with indel evolution. *BMC Bioinformatics*, **19**, 331–338.
5. Subramanian,A.R., Weyer-Menkhoff,J., Kaufmann,M. and Morgenstern,B. (2005) DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, **6**, 66–78.
6. Katoh,K., Misawa,K., ichi Kuma,K. and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.
7. Katoh,K. and Standley,D.M. (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, **30**, 772–780.
8. Felsenstein,J., Sawyer,S. and Kochin,R. (1982) An efficient method for matching nucleic acid sequences. *Nucleic Acids Res.*, **10**, 133–139.
9. Grantham,R. (1974) Amino acid difference formula to help explain protein evolution. *Science*, **185**, 862–864.
10. Cooley,J.W. and Tukey,J.W. (1965) An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, **19**, 297–301.
11. Gnann,V. and Becker,J. (2012) Signal Reconstruction from Multiresolution STFT Magnitudes with Mutual Initialization. *J. Audio Engineering Soc.*, **2012**, 274–279.
12. Stankovic,L. (2016) On the STFT inversion redundancy. *IEEE Trans. Circ. Syst. II: Express Briefs*, **63**, 284–288.
13. Bang-Jensen,J. and Gutin,G. (2009) Digraphs: theory, algorithms, and applications. *Springer Monographs in Mathematics*. 2nd edn. Springer-Verlag, London.
14. Shimbel,A. (1954) Structure in communication nets. *Proceedings of the Symposium on Information Networks*. Polytechnic Institute of Brooklyn, pp. 119–203.
15. Löytynoja,A. and Goldman,N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**, 1632–1635.
16. Nawab,S. and Dorken,E. (1993) Efficient STFT approximation using a quantization and differencing method. In: *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, Vol. **3**, pp. 587–590.
17. Nawab,S. and Dorken,E. (1995) A framework for quality versus efficiency tradeoffs in STFT analysis. *IEEE T. Signal Proces.*, **43**, 998–1001.
18. Winograd,J. and Nawab,S. (1995) Incremental refinement of DFT and STFT approximations. *IEEE Signal Proc. Let.*, **2**, 25–27.
19. Bahr,A. (2001) BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res.*, **29**, 323–326.
20. Abram,M.E., Ferris,A.L., Shao,W., Alvord,W.G. and Hughes,S.H. (2010) Nature, Position, and Frequency of Mutations Made in a Single Cycle of HIV-1 Replication. *J. Virol.*, **84**, 9864–9878.